# DEVICE AND METHOD FOR GENERATING EXPRESSION DATA IN OPERATION OF FINITE FIELD

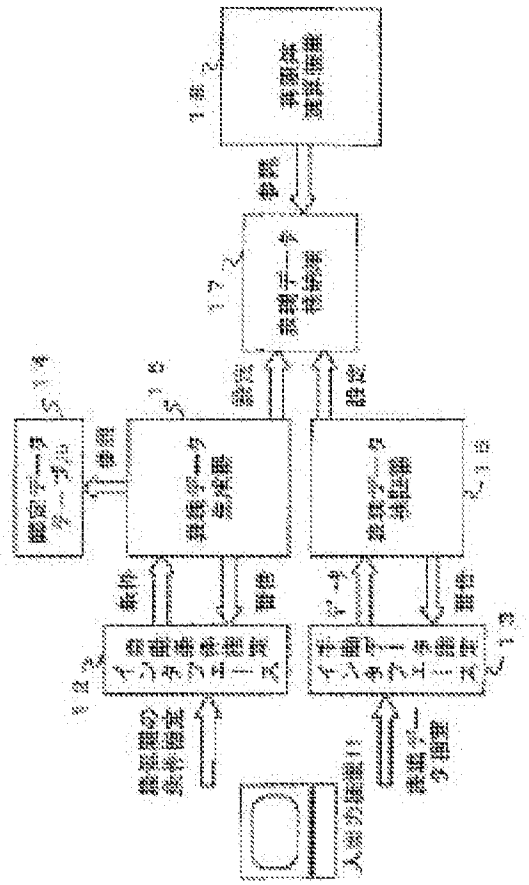
**Publication number:** JP2000293507 (A)
**Publication date:** 2000-10-20
**Inventor(s):** KOGURE ATSUSHI
**Applicant(s):** FUJITSU LTD
**Classification:**
**- international:** ***G09C1/00; G06F7/72; G06F11/10; G06F17/10; H03M13/01;*** **G09C1/00; G06F7/60; G06F11/10; G06F17/10; H03M13/00;** (IPC1-7): G06F17/10; G06F7/72; G06F11/10; G09C1/00; H03M13/01
**- European:** G06F7/72F
**Application number:** JP19990102920 19990409
**Priority number(s):** JP19990102920 19990409

**Also published as:**

- JP3833412 (B2)
- EP1043654 (A2)
- EP1043654 (A3)
- US7142668 (B1)
- CA2298995 (A1)

Abstract of **JP 2000293507 (A)**

PROBLEM TO BE SOLVED: To flexibly set up the expression data of a finite field while reducing user's load as less as possible. SOLUTION: When a user inputs a specification condition from an I/O device 11 through an automatic condition specification interface 12, an expression data generator 15 refers to a fixed data table 14, and when there is no expression data satisfying the condition in the table 14, automatically generates expression data. The generated expression data are set up in an expression data storing area 17 and a finite field arithmetic unit 18 refers to the expression data and computes a finite field.



(19)日本国特許庁（ＪＰ）　(12)**公開特許公報**（Ａ）　(11)特許出願公開番号
特開2000－293507
（P2000－293507A）
(43)公開日　平成12年10月20日(2000.10.20)

| (51)Int.Cl.⁷ | 識別記号 | ＦＩ | | テーマコード（参考） |
|---|---|---|---|---|
| Ｇ０６Ｆ　17/10 | | Ｃ０６Ｆ　15/31 | Ｚ | ５Ｂ００１ |
| 7/72 | | 7/72 | | ５Ｂ０５６ |
| 11/10 | ３３０ | 11/10 | ３３０Ｑ | ５Ｊ０６５ |
| Ｇ０９Ｃ　1/00 | ６５０ | Ｇ０９Ｃ　1/00 | ６５０Ａ | ５Ｊ１０４ |
| Ｈ０３Ｍ　13/01 | | Ｈ０３Ｍ　13/01 | | ９Ａ００１ |

審査請求　未請求　請求項の数９　ＯＬ　（全 13 頁）

(21)出願番号　特願平11－102920

(22)出願日　平成11年４月９日(1999.4.9)

(71)出願人　000005223
富士通株式会社
神奈川県川崎市中原区上小田中４丁目１番１号

(72)発明者　小葦　淳
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

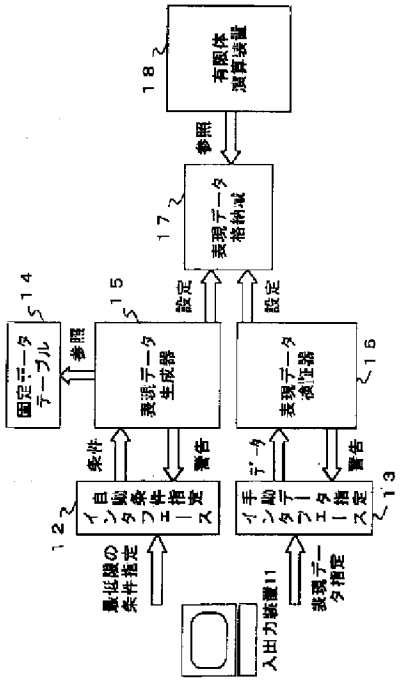
(74)代理人　100074099
弁理士　大菅　義之　（外１名）

Ｆターム(参考)　5B001 AC01
5B056 AA00 AA04 BB01 BB74 HH00
5J065 AC01
5J104 AA22 NA08 NA18 NA27
9A001 EZ03 GG17

(54)【発明の名称】　有限体演算における表現データ生成装置および方法

(57)【要約】
【課題】　ユーザの負荷をなるべく軽減しながら、有限体の表現データを柔軟に設定することが課題である。
【解決手段】　ユーザが入出力装置１１から自動条件指定インタフェース１２を介して指定条件を入力すると、表現データ生成器１５は、固定データテーブル１４を参照し、固定データテーブル１４に条件を満たす表現データがなければ、自動的に表現データを生成する。生成された表現データは表現データ格納域１７に設定され、有限体演算装置１８は、その表現データを参照して、有限体演算を行う。

第１のデータ生成装置の構成図



【特許請求の範囲】
【請求項１】　有限体を指定する条件を入力する入力手段と、
入力された条件に基づいて、前記有限体の表現データを自動的に生成する生成手段と、
生成された表現データを格納する表現データ格納手段とを備えることを特徴とするデータ生成装置。
【請求項２】　前記表現データ格納手段に格納された表現データに基づいて有限体演算を行う演算手段をさらに備えることを特徴とする請求項１記載のデータ生成装置。
【請求項３】　前記生成手段は、前記条件として前記有限体を記述する素数のビット長が入力されたとき、該ビット長に対応する素数データを自動的に生成し、前記表現データ格納手段に格納することを特徴とする請求項１記載のデータ生成装置。
【請求項４】　前記生成手段は、前記条件として前記有限体を記述する拡大次数が入力されたとき、該拡大次数に対応する既約多項式データを自動的に生成し、前記表現データ格納手段に格納することを特徴とする請求項１記載のデータ生成装置。
【請求項５】　前記生成手段は、最適化正規基底を使用する指示が入力されたとき、前記拡大次数に対応する最適化正規基底用既約多項式データを自動的に生成し、前記表現データ格納手段に格納することを特徴とする請求項４記載のデータ生成装置。
【請求項６】　あらかじめ決められた１つ以上の有限体の表現データを格納する固定データ格納手段をさらに備え、前記生成手段は、該固定データ格納手段に前記条件に対応する有限体の表現データが存在するとき、該条件に対応する有限体の表現データを前記表現データ格納手段に格納し、該固定データ格納手段に前記条件に対応する有限体の表現データが存在しないとき、該条件に対応する有限体の表現データを自動的に生成することを特徴とする請求項１記載のデータ生成装置。
【請求項７】　有限体の表現データを指定する指定手段と、指定された表現データが適切か否かを検証する検証手段とをさらに備え、該検証手段は、該指定された表現データが適切であれば、該指定された表現データを前記表現データ格納手段に格納し、該指定された表現データが適切でなければ、該指定手段に他の表現データを要求することを特徴とする請求項１記載のデータ生成装置。
【請求項８】　コンピュータのためのプログラムを記録した記録媒体であって、
有限体を指定する条件が入力されたとき、入力された条件に基づいて、該有限体の表現データを自動的に生成するステップと、
生成された表現データを出力するステップとを含む処理を前記コンピュータに実行させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。
【請求項９】　有限体を指定する条件を指定し、
指定された条件に基づいて、前記有限体の表現データを自動的に生成し、
生成された表現データを有限体演算装置に供給することを特徴とするデータ生成方法。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は、符号、暗号等に関する有限体演算を行う際に、有限体の表現データを自動的に設定するデータ生成装置およびその方法に関する。
【０００２】
【従来の技術】近年のコンピュータ技術およびネットワーク技術の発展に伴い、符号、暗号等を含む様々な分野において、有限体演算を行う局面が増えてきている。有限体（finite field）とは、四則演算が定義されている有限集合を指し、有限体演算とは、有限体上で定義されたこれらの演算を指す。
【０００３】特に、電子商取引等のインターネットを利用したビジネスにおいては、オープンネットワークのセキュリティを確保するための暗号理論が非常に重要な技術として脚光を浴びている。暗号技術を実現するためには有限体演算が必須であるが、データのセキュリティを保つために非常に大きなサイズの有限体を用いるケースが多く、その演算効率を上げることは、実用上重要なテーマである。
【０００４】有限体演算を実現するためには、まず、いくつかの表現データを決定して、有限体を表現しなければならない。有限体の要素（元）の個数は、ｐを素数とし、ｍを正整数として、$p^m$ となる。この正整数ｍは拡大次数と呼ばれる。一般に、有限体の要素を記述するためには、素数ｐ、拡大次数ｍ、およびｍ次の既約多項式Ｆ（ｘ）を表現データとして設定しなければならないことが知られている。
【０００５】従来の有限体演算装置においては、装置自身があらかじめ内蔵している固定の表現データを用いるか、もしくは、ユーザがすべての表現データを指定することで、有限体を表現していた。
【０００６】
【発明が解決しようとする課題】しかしながら、従来の有限体の表現方法には、次のような問題がある。固定の表現データのみを用いる場合、符号、暗号等のシステムにおける様々な局面に必要となる多様な表現データを得ることができず、表現の柔軟性に欠ける。また、非常に大きな有限体において、ユーザがすべての表現データを指定する場合、表現データの選択に多大な労力を要する。さらに、ユーザが不適切な表現データを選択すると、演算速度が低下し、システムの性能が低下するといった弊害が生じることもある。
【０００７】本発明の課題は、効率の良い有限体演算を実現するために、ユーザの負荷をなるべく軽減しなが

ら、有限体の表現データを柔軟に設定するデータ生成装置およびその方法を提供することである。

【０００８】

【課題を解決するための手段】図１は、本発明のデータ生成装置の原理図である。図１のデータ生成装置は、入力手段１、生成手段２、および表現データ格納手段３を備える。入力手段１は、有限体を指定する条件を入力する。生成手段２は、入力された条件に基づいて、有限体の表現データを自動的に生成し、表現データ格納手段３は、生成された表現データを格納する。

【０００９】生成手段２は、入力手段１が入力した指定条件に基づいて、その条件に対応する有限体の表現データを自動的に生成し、生成された表現データを表現データ格納手段３に格納する。格納された表現データは、有限体演算を行う装置に供給され、その表現データを用いた有限体演算が行われる。

【００１０】ユーザは、入力手段１を介して、希望する有限体の条件を入力することができる。このとき、例えば、素数ｐのビット長や拡大次数ｍのような、直観的に思いつくことのできる最小限の条件さえ指定すれば、生成手段２が、素数ｐやｍ次の既約多項式Ｆ（ｘ）のような、直観的に思いつくことの困難な表現データを自動的に生成する。

【００１１】したがって、有限体の表現データの選択におけるユーザの負荷が軽減され、生成手段２が生成可能な範囲内で、表現データを柔軟に設定することが可能になる。さらに、適切な条件が指定されれば、効率の良い有限体演算を実現する表現データが生成される。このように、本発明のポイントは、指定された条件に基づいて、有限体の表現データを自動的に生成することである。

【００１２】例えば、図１の入力手段１は、後述する図４の入出力装置１１および自動条件指定インタフェース１２に対応し、図１の生成手段２は、図４の表現データ生成器１５に対応し、図１の表現データ格納手段３は、図４の表現データ格納域１７に対応する。

$$s=(g^{y})^{x}=9^{4}=5 \pmod{p} \quad (1)$$

また、ユーザＢは、ユーザＡから受信した$g^{x} \bmod p=5$を用いて、次式により共有秘密情報ｓを求める。

$$s=(g^{x})^{y}=5^{6}=5 \pmod{p} \quad (2)$$

ここでは、簡単のためにｐ＝１１としたが、実際には、ｐとして非常に大きな値が用いられる。ｐが十分に大きければ、たとえ第３者が$g^{x} \bmod p$と$g^{y} \bmod p$を取得したとしても、これらの情報からｘとｙを求めることは極めて困難となり、秘密情報ｓを知ることは事実上不可能である。こうして、ユーザＡとユーザＢは、秘密情報ｓを安全に共有することができ、この情報を秘密鍵として用いて暗号通信を行うことができる。

【００１３】

【発明の実施の形態】以下、図面を参照しながら、本発明の実施の形態を詳細に説明する。まず、図２および図３を参照しながら、暗号技術に用いられる有限体演算の例を説明する。

【００１４】図２は、よく知られたDiffie-Hellmanの鍵交換の例を示している。例えば、ユーザＡとユーザＢの間で秘密鍵暗号を用いてデータ通信を行う場合、まず、両者が秘密鍵を共有する必要がある。このとき、第３者に知られることなく、ネットワーク上で秘密情報を交換するために、Diffie-Hellmanの鍵交換が用いられる。

【００１５】素数ｐ、拡大次数ｍにより表される要素数$p^{m}$の有限体をＧＦ（$p^{m}$）と記すことにすると、図２で用いられている有限体は、ＧＦ（１１）と記される（ｍ＝１）。ｍ＝１の有限体は、素体（prime field）とも呼ばれる。

【００１６】ここでは、ｐ＝１１、ｇ＝２が公開されており、ユーザＡの秘密情報がｘ＝４であり、ユーザＢの秘密情報がｙ＝６であり、共有すべき秘密情報が$s=g^{xy} \bmod p=2^{24} \bmod 11=5$である場合を考える。ｍｏｄ　ｐは、ｐを法とする剰余演算を表し、$g^{xy} \bmod p$は、ＧＦ（１１）における有限体演算の一例を表している。

【００１７】このとき、ユーザＡは、$g^{x} \bmod p=2^{4} \bmod 11=5$をユーザＢに送信し、ユーザＢは、$g^{y} \bmod p=2^{6} \bmod 11=9$をユーザＡに送信する。次に、ユーザＡは、ユーザＢから受信した$g^{y} \bmod p=9$を用いて、次式により共有秘密情報ｓを求める。

【００１８】次に、図３は、Diffie-Hellmanの鍵交換において、有限体ＧＦ（２）＝｛０，１｝の要素（０または１）を係数とする３次既約多項式を用いて表される有限体ＧＦ（$2^{3}$）を用いた例を示している。この場合、ＧＦ（$2^{3}$）は、ＧＦ（２）の要素を係数とする項からなる$2^{3}$個の２次以下の多項式を要素として持つ。

【００１９】ここでは、ｐ＝２、３次既約多項式ｆ（ｘ）＝$x^{3}+x+1$、ｇ＝ｘ　ｍｏｄｆ（ｘ）が公開されており、ユーザＡの秘密情報がｉ＝４であり、ユーザＢの秘密情報がｊ＝６であり、共有すべき秘密情報が$s=g^{ij} \bmod f(x)$である場合を考える。ｍｏｄ　ｆ（ｘ）は、ｆ（ｘ）を法とする多項式の剰余演算を表し、$g^{ij} \bmod f(x)$は、ＧＦ（$2^{3}$）における有限体演算の一例を表している。

【００２０】このとき、ユーザＡは、$g^{i} \bmod f(x)=x^{4} \bmod (x^{3}+x+1)=x^{2}+x$をユーザＢに送信し、ユーザＢは、$g^{j} \bmod f(x)=x^{6} \bmod (x^{3}+x+1)=x^{2}+1$をユーザＡに送信する。次に、ユーザＡは、ユーザＢから受信した$g^{j} \bmod f(x)=x^{2}+1$を用いて、次式により共

有秘密情報sを求める。

$$s=(g^j)^i=(x^2+1)^4=x+1 \pmod{f(x)} \quad (3)$$

また、ユーザBは、ユーザAから受信した$g^i \bmod f(x)$を用いて、次式により共有秘密情報sを求める。

$$s=(g^i)^j=(x^2+x)^6=x+1 \pmod{f(x)} \quad (4)$$

ここでは、簡単のためにm＝3としたが、実際には、mとして非常に大きな値が用いられる。mが十分に大きければ、たとえ第3者が$g^i \bmod f(x)$と$g^j \bmod f(x)$を取得したとしても、これらの情報からiとjを求めることは極めて困難となり、第3者が秘密情報sを知ることは事実上不可能である。こうして、ユーザAとユーザBは、秘密情報sを安全に共有することができ、この情報を秘密鍵として用いて暗号通信を行うことができる。

【0021】上述したように、素体GF(p)を表現するためには、素数pを指定しなければならず、有限体GF($2^m$)を表現するためには、拡大次数mとm次の既約多項式を指定しなければならない。しかしながら、pやmとして非常に大きな数を用いる場合、ユーザが素数pやm次の既約多項式を陽に指定することは難しい。

【0022】本実施形態のデータ生成装置は、ユーザが直観的に思いつくことのできる最小限の条件さえ指定すれば、自動的に素数pやm次の既約多項式等の表現データを生成する。このため、ユーザが表現データを指定する際の負荷が大きく軽減される。

【0023】このとき、可能であれば、高速演算に適した表現データを選択することで、より高速な有限体演算が実現される。さらに、ユーザがすべての表現データを指定することのできるインタフェースを設けることにより、様々な局面に対応できる柔軟性が実現される。

【0024】図4は、このようなデータ生成装置の基本構成図である。図4のデータ生成装置は、入出力装置11、自動条件指定インタフェース12、手動データ指定インタフェース13、固定データテーブル14、表現データ生成器15、表現データ検証器16、および表現データ格納域17を備え、有限体演算装置18が用いる有限体の表現データを生成する。

【0025】ユーザは、入出力装置11を介して、自動条件指定インタフェース12または手動データ指定インタフェース13に対して必要な指示を入力する。自動条件指定インタフェース12は、ユーザが表現データ生成のための最小限の条件を指定する場合に用いられ、手動データ指定インタフェース13は、ユーザが表現データの任意のパラメータを自ら指定する場合に用いられる。

【0026】固定データテーブル14は、あらかじめ決められた適切な表現データを格納し、表現データ生成器15は、自動条件指定インタフェース12による指定に従って、最終的に使用される表現データを決定する。表現データ検証器16は、手動データ指定インタフェース13により指定された表現データが有限体演算に適切なものかどうかを検証する。

【0027】表現データ格納域17は、実際の演算時に使用される表現データを格納し、有限体演算装置18は、表現データ格納域17の表現データを参照して、有限体演算を行う。

【0028】図5は、図4のデータ生成装置による処理のフローチャートである。ユーザは表現データ選択のための条件指定において、自動モードまたは手動モードのいずれかを選択することができ、データ生成装置は、自動条件指定インタフェース12および手動データ指定インタフェース13により、どちらのモードが選択されたかをチェックする（ステップS1）。自動モードにおいては、ユーザは最低限の条件を指定するだけで済み、手動モードにおいては、ユーザは表現データのすべてのパラメータを指定しなければならない。

【0029】自動モードが選択された場合、表現データ生成器15は、まず、指定された条件が適切か否か（正当か否か）をチェックする（ステップS2）。そして、それが適切でなければ、自動条件指定インタフェース12を介して警告を発し、条件の再入力を要求して（ステップS3）、ステップS2以降の処理を繰り返す。

【0030】適切な条件が入力されると、次に、固定データテーブル14を検索して、指定された条件を満たす表現データが存在するか否かをチェックする（ステップS4）。そのような表現データが固定データテーブル14に存在すれば、そのパラメータを表現データ格納域17に設定し（ステップS6）、処理を終了する。また、そのような表現データが存在しなければ、指定された条件に基づいて表現データを生成して（ステップS5）、表現データ格納域17に設定し（ステップS6）、処理を終了する。

【0031】固定データテーブル14にあらかじめ保持された表現データは、例えば、高速な有限体演算を実現できるように選ばれたものであり、表現データ生成器15による表現データ生成のロジックは、例えば、高速な有限体演算を実現する表現データを生成するものである。

【0032】ステップS1において手動モードが選択された場合、表現データ検証器16は、指定された表現データの正当性を検証し（ステップS7）、それが有限体演算を行うにあたって適切なものかどうかをチェックする（ステップS8）。

【0033】そして、その表現データが適切であれば、それを表現データ格納域17に設定し（ステップS6）、処理を終了する。また、指定された表現データが

適切でなければ、手動データ指定インタフェース１３を介して警告を発し、再度、適切な表現データを指定するようにユーザに促し（ステップＳ９）、ステップＳ７以降の処理を繰り返す。

【００３４】表現データが表現データ格納域１７に設定されると、有限体演算部１８は、その値に基づいて有限体演算を行う。自動モードが選択された場合、自動的に高速な有限体演算を実現する表現データが設定されるため、演算の効率が向上する。

【００３５】次に、図６から図１２までを参照しながら、図４の構成に基づく様々なデータ生成装置の実施形態について説明する。図６は、素数ｐに関する条件に基づいて素体ＧＦ（ｐ）の表現データを生成するデータ生成装置の構成図である。素体ＧＦ（ｐ）の表現データとしては、素数ｐを設定すればよい。この数ｐは、標数（characteristic）とも呼ばれる。

【００３６】図６のデータ生成装置は、入出力装置１１、自動条件指定インタフェース１２、手動データ指定インタフェース１３、素数表２１、テーブル参照部２２、乱数発生部２３、素数判定部２４、２５、および素数格納域２６を備え、有限体演算装置１８が用いる素体の表現データを生成する。

【００３７】素数表２１は、図４の固定データテーブル１４に対応し、あらかじめ決められた素数とそのバイナリコードのビット長ｎの対応関係を格納する。例えば、図７に示す素数表においては、ｎ＝２，３，４，．．．に対応する素数３，７，１１，．．．が順に格納されている。

【００３８】テーブル参照部２２、乱数発生部２３、および素数判定部２４は、図４の表現データ生成器１５に対応する。テーブル参照部２２は、与えられたビット長ｎをキーとして素数表２１を参照し、対応する素数を取得する。与えられたビット長ｎに対応する素数が素数表２１に存在しないとき、乱数発生部２３は、ビット長ｎの乱数を発生し、素数判定部２４は、その乱数が素数か否かを判定する。

【００３９】素数判定部２５は、図４の表現データ検証器１６に対応し、ユーザが指定した素数ｐが本当に素数であるか否かを判定する。素数格納域２６は、図４の表現データ格納域１７に対応し、与えられた素数ｐを表現データとして格納する。

【００４０】図８は、図６のデータ生成装置による処理のフローチャートである。まず、データ生成装置は、自動モードと手動モードのいずれが選択されたかをチェックする（ステップＳ１１）。自動モードにおいては、ユーザは自動条件指定インタフェース１２を介して素数のビット長ｎ（素体のビットサイズ）のみを入力し、手動モードにおいては、ユーザは自ら具体的な素数ｐを選択し、手動データ指定インタフェース１３を介してそれを入力する。

【００４１】自動モードが選択された場合、テーブル参照部２２は、指定されたビット長ｎがサポート範囲内のビット長か否かをチェックする（ステップＳ１２）。そして、それがサポート範囲外であれば、自動条件指定インタフェース１２を介して警告を発し、ビット長の再入力を要求して（ステップＳ１３）、ステップＳ１２以降の処理を繰り返す。

【００４２】サポート範囲内のビット長ｎが入力されると、次に、素数表２１を検索して、ビット長ｎに対応する素数が存在するか否かをチェックする（ステップＳ１４）。そのような素数が素数表２１に存在すれば、それを素数ｐとして素数格納域２６に設定し（ステップＳ１６）、処理を終了する。

【００４３】また、そのような素数が存在しなければ、乱数発生部２３は、指定されたビット長ｎの乱数を発生させ、素数判定部２４は、発生した乱数が素数か否かを判定する（ステップＳ１５）。そして、素数判定部２４は、素数と判定された乱数を表現データとして採用し、それを素数ｐとして素数格納域２６に設定して（ステップＳ１６）、処理を終了する。

【００４４】素数判定部２４による素数判定（primality test）のアルゴリズムとしては、例えば、以下の文献に記載されているようなものが用いられる。

文献［１］：情報処理学会　監修，岡本龍明（おかもと　たつあき）・太田和夫（おおた　かずお）　共編，“暗号・ゼロ知識証明・数論”，共立出版，ｐｐ．１３０－１４３，１９９５．

文献［２］：IEEE P1363 Annex A/Editorial Contribution, “Standard Specifications For Public Key Cryptography ”, pp.78-81, 1998.

ステップＳ１１において手動モードが選択された場合、素数判定部２５は、素数判定部２４と同様にして、指定された数ｐの素数判定を行い（ステップＳ１７）、それが本当に素数であるか否かをチェックする（ステップＳ１８）。

【００４５】そして、指定された数ｐが素数であれば、それを表現データとして採用して、素数格納域２６に設定し（ステップＳ１６）、処理を終了する。また、それが素数でなければ、手動データ指定インタフェース１３を介して警告を発し、再度、正しい素数を入力するようにユーザに促し（ステップＳ１９）、ステップＳ１７以降の処理を繰り返す。

【００４６】素数ｐが素数格納域２６に設定されると、有限体演算部１８は、その値に基づいて素体ＧＦ（ｐ）上の有限体演算を行う。このような構成によれば、ユーザがビット長さえ指定すれば自動的に素数ｐが生成されるため、ユーザが素数ｐを指定する際の負荷が軽減される。また、必要であれば、ユーザが素数ｐを直接指定することもでき、柔軟な操作性が実現される。

【００４７】図９は、拡大次数ｍに基づいて有限体ＧＦ

（$2^m$）の表現データを生成するデータ生成装置の構成図である。有限体ＧＦ（$2^m$）の場合、素数ｐは２に決められているため、それ以外の表現データとしては、拡大次数ｍとｍ次の既約多項式Ｆ（ｘ）を設定すればよい。ｍ次の多項式は、例えば、各項$x^k$（ｋ＝０，１，．．．，ｍ）の係数（０または１）の集合により表すことができる。

【００４８】図９のデータ生成装置は、入出力装置１１、自動条件指定インタフェース１２、手動データ指定インタフェース１３、既約多項式表３１、テーブル参照部３２、多項式発生部３３、既約性判定部３４、３５、および既約多項式格納域３６を備え、有限体演算装置１８が用いる有限体の表現データを生成する。

【００４９】既約多項式表３１は、図４の固定データテーブル１４に対応し、あらかじめ決められた拡大次数ｍとｍ次の既約多項式Ｆ（ｘ）の対応関係を格納する。例えば、図１０に示す既約多項式表においては、ｍ＝２，３，４，．．．に対応する既約多項式として、$x^2+x+1$，$x^3+x+1$，$x^4+x+1$，．．．が格納されている。

【００５０】テーブル参照部３２、多項式発生部３３、および既約性判定部３４は、図４の表現データ生成器１５に対応する。テーブル参照部３２は、与えられた拡大次数ｍをキーとして既約多項式表３１を参照し、対応する既約多項式を取得する。与えられた拡大次数ｍに対応する既約多項式が既約多項式表３１に存在しないとき、多項式発生部３３は、ｍ次の多項式を発生し、既約性判定部３４は、その多項式が既約か否かを判定する。

【００５１】既約性判定部３５は、図４の表現データ検証器１６に対応し、ユーザが指定したｍ次の既約多項式が本当に既約であるか否かを判定する。既約多項式格納域３６は、図４の表現データ格納域１７に対応し、拡大次数ｍとｍ次の既約多項式Ｆ（ｘ）を表現データとして格納する。

【００５２】図１１は、図９のデータ生成装置による処理のフローチャートである。まず、データ生成装置は、自動モードと手動モードのいずれが選択されたかをチェックする（ステップＳ２１）。自動モードにおいては、ユーザは自動条件指定インタフェース１２を介して拡大次数ｍのみを入力し、手動モードにおいては、ユーザは自ら具体的なｍ次の既約多項式Ｆ（ｘ）を選択し、それを拡大次数ｍとともに手動データ指定インタフェース１３を介して入力する。

【００５３】自動モードが選択された場合、テーブル参照部３２は、指定された拡大次数ｍがサポート範囲内の拡大次数か否かをチェックする（ステップＳ２２）。そして、それがサポート範囲外であれば、自動条件指定インタフェース１２を介して警告を発し、拡大次数の再入力を要求して（ステップＳ２３）、ステップＳ２２以降の処理を繰り返す。

【００５４】サポート範囲内の拡大次数ｍが入力されると、次に、既約多項式表３１を検索して、拡大次数ｍに対応する既約多項式が存在するか否かをチェックする（ステップＳ２４）。そのような既約多項式が既約多項式表３１に存在すれば、それを既約多項式Ｆ（ｘ）として、拡大次数ｍとともに既約多項式格納域３６に設定し（ステップＳ２６）、処理を終了する。

【００５５】また、そのような既約多項式が存在しなければ、多項式発生部３３は、ｍ次のの多項式を発生させ、既約性判定部３４は、発生した多項式が既約か否かを判定する（ステップＳ２５）。そして、既約性判定部３４は、既約と判定された多項式を既約多項式Ｆ（ｘ）として採用し、それを拡大次数ｍとともに既約多項式格納域３６に設定して（ステップＳ２６）、処理を終了する。

【００５６】ステップＳ２１において手動モードが選択された場合、既約性判定部３５は、既約性判定部３４と同様にして、指定された多項式Ｆ（ｘ）の既約性判定（irreducibility test ）を行い（ステップＳ２７）、それが本当に既約であるか否かをチェックする（ステップＳ２８）。

【００５７】そして、指定された多項式Ｆ（ｘ）が既約であれば、それを表現データとして採用して、拡大次数ｍとともに既約多項式格納域３６に設定し（ステップＳ２６）、処理を終了する。また、それが既約でなければ、手動データ指定インタフェース１３を介して警告を発し、再度、正しい既約多項式を入力するようにユーザに促し（ステップＳ２９）、ステップＳ２７以降の処理を繰り返す。

【００５８】図１２は、図１１のステップＳ２５における既約多項式生成処理のフローチャートである。まず、テーブル参照部３２は、多項式発生部３３に拡大次数ｍを入力し（ステップＳ３１）、多項式発生部３３は、（ｍ＋１）ビットの乱数（バイナリコード）を発生させる（ステップＳ３２）。

【００５９】次に、多項式発生部３３は、有限体ＧＦ（２）＝｛０，１｝の要素を係数とするｍ次の多項式を生成する（ステップＳ３３）。ここでは、ｋ＝１，２，．．．，ｍ＋１のそれぞれの値について、発生した乱数のｋビット目の値（０または１）を$x^{k-1}$の項の係数とし、それらの項からなるｍ次の多項式を生成する。

【００６０】次に、既約性判定部３４は、生成された多項式の既約性判定を行い（ステップＳ３４）、それが既約か否かをチェックする（ステップＳ３５）。既約性判定のアルゴリズムとしては、例えば、上述した文献［２］の３０ページに記載されているようなものが用いられる。生成された多項式が既約でなければ、ステップＳ３２以降の処理を繰り返し、既約な多項式が得られれば、それを既約多項式格納域３６に出力して（ステップＳ３６）、処理を終了する。

【００６１】図１１の処理により、拡大次数mと既約多項式Ｆ（ｘ）が既約多項式格納域３６に設定されると、有限体演算部１８は、そのデータに基づいて有限体ＧＦ（$2^m$）上の有限体演算を行う。このような構成によれば、ユーザが拡大次数mさえ指定すれば自動的にm次の既約多項式Ｆ（ｘ）が生成されるため、ユーザが既約多項式Ｆ（ｘ）を指定する際の負荷が軽減される。また、必要であれば、ユーザが既約多項式Ｆ（ｘ）を直接指定することもでき、柔軟な操作性が実現される。

【００６２】ところで、有限体ＧＦ（$2^m$）上の演算を高速化する方法の１つとして、最適化正規基底（optimal normal basis）を用いる方法が知られている（文献［１］，ｐｐ．１６７－１７０／文献［２］，ｐｐ．１９－２４）。一般に、有限体ＧＦ（$2^m$）は、正規基底（normal basis）を生成元として表現することができ、特に、最適化正規基底に基づく表現データを用いれば、べき乗算および乗算を飛躍的に高速化することができる。

【００６３】そこで、図９のデータ生成装置において、最適化正規基底を用いた表現データを生成するオプションを設けることにする。ただし、最適化正規基底は、必ずしもすべての拡大次数について存在するわけではなく、これを利用するためには、拡大次数mを適切に指定する必要がある。例えば、暗号に用いられる１６０≦m≦２０００の拡大次数のうち最適化正規基底が存在するものは、以下の文献にリストアップされている。

文献［３］：American National Standard, x9.62-199x, Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm, p.41, Working Draft.

ユーザは、自動条件指定インタフェース１２および手動データ指定インタフェース１３を介して、最適化正規基底を使用するか否かをオプションとして指定することができる。この場合、既約多項式表３１には、各拡大次数について、最適化正規基底が存在するか否かを表す情報が付加され、存在する場合は、必要に応じてその最適化正規基底用既約多項式の情報が付加される。最適化正規基底用既約多項式とは、最適化正規基底を用いて有限体を表現するために必要な特定の既約多項式を指す。

【００６４】図１３は、このようなデータ生成装置による処理のフローチャートである。まず、データ生成装置は、自動モードと手動モードのいずれが選択されたかをチェックする（ステップＳ４１）。自動モードにおいては、ユーザは、自動条件指定インタフェース１２を介して、拡大次数mと最適化正規基底を使用する指示とを入力する。また、手動モードにおいては、ユーザは、自ら具体的なm次の最適化正規基底用既約多項式Ｆ（ｘ）を選択し、それを拡大次数mとともに手動データ指定インタフェース１３を介して入力する。

【００６５】自動モードが選択された場合、テーブル参照部３２は、既約多項式表３１を参照して、指定された拡大次数mの最適化正規基底が存在するか否かをチェックする（ステップＳ４２）。そして、m次の最適化正規基底が存在しないか、あるいは拡大次数mがサポート範囲外であれば、自動条件指定インタフェース１２を介して警告を発し、拡大次数の再入力を要求して（ステップＳ４３）、ステップＳ４２以降の処理を繰り返す。

【００６６】最適化正規基底を有する拡大次数mが入力されると、次に、既約多項式表３１に最適化正規基底用既約多項式が存在するか否かをチェックする（ステップＳ４４）。そのような既約多項式が既約多項式表３１に存在すれば、それを最適化正規基底用既約多項式Ｆ（ｘ）として、拡大次数mとともに既約多項式格納域３６に設定し（ステップＳ４６）、処理を終了する。

【００６７】また、そのような既約多項式が存在しなければ、多項式発生部３３は、m次の最適化正規基底用既約多項式を生成する（ステップＳ４５）。そして、それを最適化正規基底用既約多項式Ｆ（ｘ）として、拡大次数mとともに既約多項式格納域３６に設定し（ステップＳ４６）、処理を終了する。最適化正規基底用既約多項式の生成アルゴリズムとしては、例えば、上述した文献［２］の３７ページに記載されているようなものが用いられる。

【００６８】ステップＳ４１において手動モードが選択された場合、既約性判定部３５は、指定された多項式Ｆ（ｘ）の最適化正規基底用既約性判定を行い（ステップＳ４７）、それが本当に最適化正規基底用既約多項式であるか否かをチェックする（ステップＳ４８）。

【００６９】最適化正規基底用既約性判定は、例えば、上述した文献［２］の２０ページに記載されている最適化正規基底が存在するか否かの判定アルゴリズムと、最適化正規基底用既約多項式の生成アルゴリズムとを組み合わせて行われる。

【００７０】この場合、既約性判定部３５は、まず、指定された多項式Ｆ（ｘ）の拡大次数mにおいて、最適化正規基底が存在するか否かをチェックする。最適化正規基底が存在すれば、次に、m次の最適化正規基底用既約多項式を生成して、Ｆ（ｘ）を生成された多項式と比較する。そして、両者が一致すれば、Ｆ（ｘ）が最適化正規基底用既約多項式であると判定する。

【００７１】また、あらかじめこれらのアルゴリズムに基づいて、サポート範囲内のすべての最適化正規基底用既約多項式のテーブルを生成しておき、そのテーブルを参照しながら、最適化正規基底用既約性判定を行うこともできる。

【００７２】指定された多項式Ｆ（ｘ）が最適化正規基底用既約多項式であれば、それを表現データとして採用して、拡大次数mとともに既約多項式格納域３６に設定し（ステップＳ４６）、処理を終了する。また、それが最適化正規基底用既約多項式でなければ、手動データ指

定インタフェース１３を介して警告を発し、再度、正しい最適化正規基底用既約多項式を入力するようにユーザに促し（ステップＳ４９）、ステップＳ４７以降の処理を繰り返す。

【００７３】拡大次数ｍと最適化正規基底用既約多項式Ｆ（ｘ）が既約多項式格納域３６に設定されると、有限体演算部１８は、そのデータに基づいて有限体ＧＦ（$2^m$）上の有限体演算を行う。

【００７４】このような構成によれば、ユーザが最適化正規基底を有する拡大次数ｍさえ指定すれば、自動的にｍ次の最適化正規基底用既約多項式Ｆ（ｘ）が生成される。このため、ユーザが最適化正規基底用既約多項式Ｆ（ｘ）を指定する際の負荷が軽減されるとともに、最適化正規基底に基づく高速な有限体演算が可能となる。また、必要であれば、ユーザが最適化正規基底用既約多項式Ｆ（ｘ）を直接指定することもでき、柔軟な操作性が実現される。

【００７５】以上の実施形態においては、主として、有限体ＧＦ（ｐ）およびＧＦ（$2^m$）の表現データを生成する場合について説明したが、一般の有限体ＧＦ（$p^m$）の表現データを生成する場合も同様である。この場合、ユーザは、素数ｐまたはそのビット長ｎと拡大次数ｍさえ指定すれば、データ生成装置は、自動的にＧＦ（$p^m$）の表現データを生成する。

【００７６】ところで、本実施形態のデータ生成装置および有限体演算装置は、図１４に示すような情報処理装置（コンピュータ）を用いて構成することができる。図１４の情報処理装置は、ＣＰＵ（中央処理装置）４１、メモリ４２、入力装置４３、出力装置４４、外部記憶装置４５、媒体駆動装置４６、およびネットワーク接続装置４７を備え、それらはバス４８により互いに接続されている。

【００７７】メモリ４２は、例えば、ＲＯＭ（read only memory）、ＲＡＭ（random access memory）等を含み、処理に用いられるプログラムとデータを格納する。ＣＰＵ４１は、メモリ４２を利用してプログラムを実行することにより、必要な処理を行う。

【００７８】この場合、図４の自動条件指定インタフェース１２、手動データ指定インタフェース１３、表現データ生成器１５、および表現データ検証器１６は、メモリ４２の特定のプログラムコードセグメントに格納されたインストラクションの集合により実現されるソフトウェアコンポーネントに対応する。また、図４の固定データテーブル１４と表現データ格納域１７は、メモリ４２内に設けられる。

【００７９】入力装置４３は、例えば、キーボード、ポインティングデバイス、タッチパネル等であり、ユーザからの指示や情報の入力に用いられる。出力装置４４は、例えば、ディスプレイ、プリンタ、スピーカ等であり、ユーザへのメッセージや処理結果の出力に用いられる。

【００８０】外部記憶装置４５は、例えば、磁気ディスク装置、光ディスク装置、光磁気ディスク（magneto-optical disk）装置等である。情報処理装置は、この外部記憶装置４５に、上述のプログラムとデータを保存しておき、必要に応じて、それらをメモリ４２にロードして使用することができる。

【００８１】媒体駆動装置４６は、可搬記録媒体４９を駆動し、その記録内容にアクセスする。可搬記録媒体４９としては、メモリカード、フロッピーディスク、ＣＤ－ＲＯＭ（compact disk read only memory ）、光ディスク、光磁気ディスク等、任意のコンピュータ読み取り可能な記録媒体が用いられる。ユーザは、この可搬記録媒体４９に上述のプログラムとデータを格納しておき、必要に応じて、それらをメモリ４２にロードして使用することができる。

【００８２】ネットワーク接続装置４７は、任意のネットワーク（回線）を介して外部の装置と通信し、通信に伴うデータ変換を行う。情報処理装置は、必要に応じて、ネットワーク接続装置４７を介して上述のプログラムとデータを外部の装置から受け取り、それらをメモリ４２にロードして使用することができる。

【００８３】図１５は、図１４の情報処理装置にプログラムとデータを供給することのできるコンピュータ読み取り可能な記録媒体を示している。可搬記録媒体４９や外部のデータベース５０に保存されたプログラムとデータは、メモリ４２にロードされる。そして、ＣＰＵ４１は、そのデータを用いてそのプログラムを実行し、必要な処理を行う。

【００８４】

【発明の効果】本発明によれば、ユーザが最小限の条件さえ指定すれば自動的に有限体の表現データが生成されるため、ユーザの負荷が大きく軽減される。また、特定の表現データを選択的に生成することにより、高速な有限体演算を実現することができる。さらに、有限体演算の様々な局面に応じて、ユーザがすべての表現データを柔軟に指定することもできる。

【図面の簡単な説明】

【図１】本発明のデータ生成装置の原理図である。

【図２】第１の秘密情報の交換を示す図である。

【図３】第２の秘密情報の交換を示す図である。

【図４】第１のデータ生成装置の構成図である。

【図５】第１のデータ生成処理のフローチャートである。

【図６】第２のデータ生成装置の構成図である。

【図７】素数表を示す図である。

【図８】第２のデータ生成処理のフローチャートである。

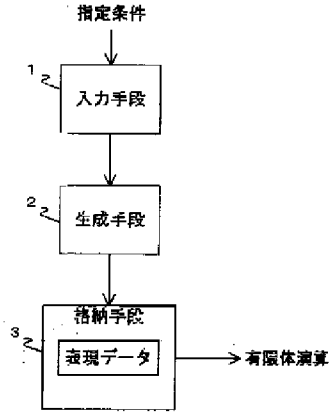
【図９】第３のデータ生成装置の構成図である。

【図１０】既約多項式表を示す図である。

【図１１】第３のデータ生成処理のフローチャートである。
【図１２】既約多項式生成処理のフローチャートである。
【図１３】第４のデータ生成処理のフローチャートである。
【図１４】情報処理装置の構成図である。
【図１５】記録媒体を示す図である。
【符号の説明】
１　入力手段
２　生成手段
３　表現データ格納手段
１１　入出力装置
１２　自動条件指定インタフェース
１３　手動データ指定インタフェース
１４　固定データテーブル
１５　表現データ生成器
１６　表現データ検証器
１７　表現データ格納域
１８　有限体演算装置
２１　素数表
２２、３２　テーブル参照部
２３　乱数発生部
２４、２５　素数判定部
２６　素数格納域
３１　既約多項式表
３３　多項式発生部
３４、３５　既約性判定部
３６　既約多項式格納域
４１　ＣＰＵ
４２　メモリ
４３　入力装置
４４　出力装置
４５　外部記憶装置
４６　媒体駆動装置
４７　ネットワーク接続装置
４８　バス
４９　可搬記録媒体
５０　データベース
//

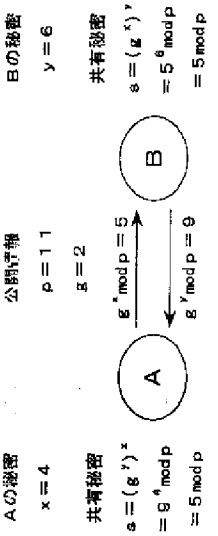
【図１】

本　発　明　の　原　理　図



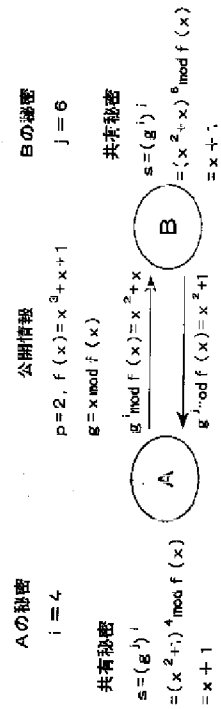

【図２】

第１の秘密情報の交換を示す図



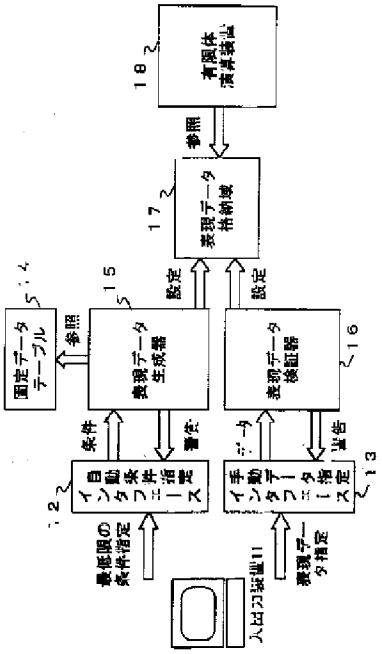
【図3】

第２の秘密情報の交換を示す図



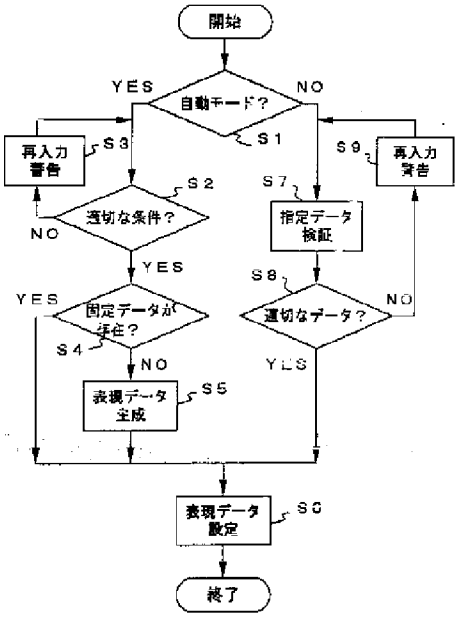

【図4】

第１のデータ生成装置の構成図



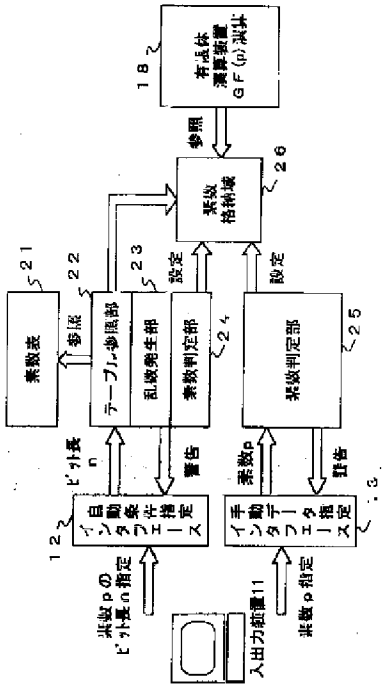

【図5】

第１のデータ生成処理のフローチャート




【図6】

第２のデータ生成装置の構成図



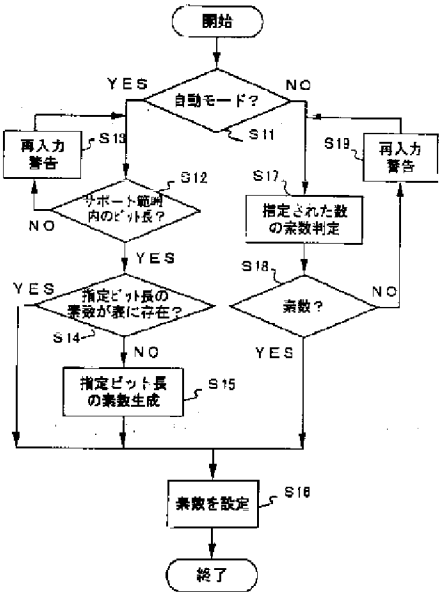
【図７】

素数表を示す図

| ビット長n | nビット素数 |
|---|---|
| 2 | 3 |
| 3 | 7 |
| 4 | 11 |
| ⋮ | ⋮ |

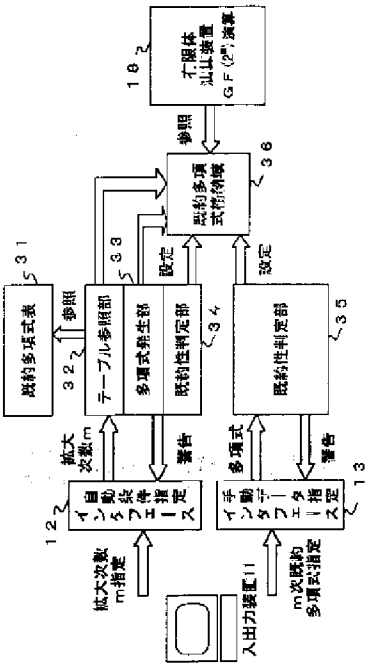
【図８】

第２のデータ生成処理のフローチャート




【図９】

第３のデータ生成装置の構成図



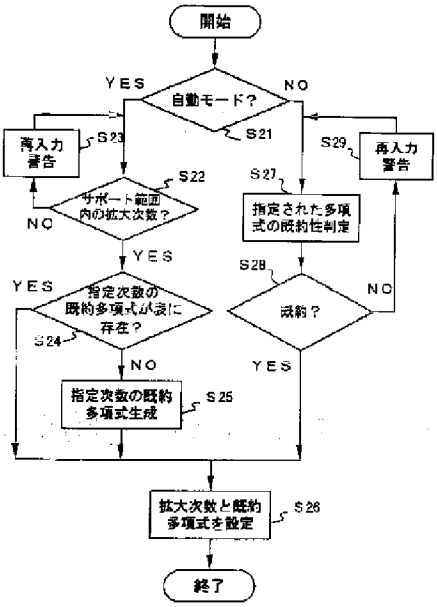

【図１０】

既約多項式表を示す図

| 拡大次数m | m次既約多項式 |
|---|---|
| 2 | $x^2+x+1$ |
| 3 | $x^3+x+1$ |
| 4 | $x^4+x+1$ |
| ⋮ | ⋮ |

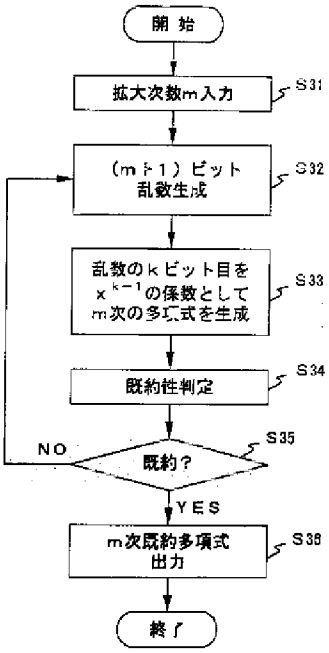
【図１１】

第３のデータ生成処理のフローチャート



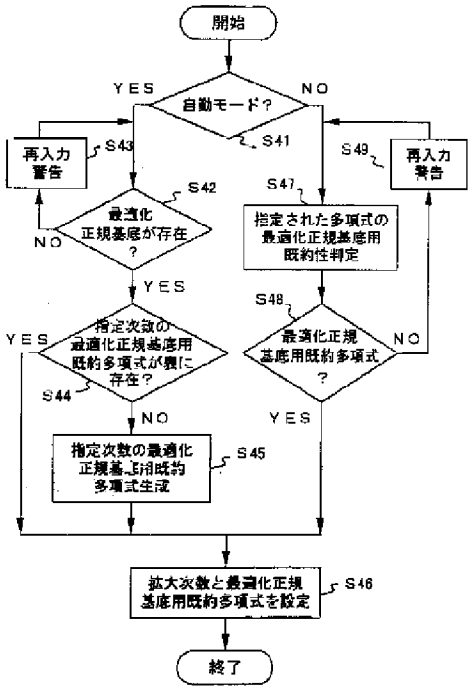

【図１２】

既約多項式生成処理のフローチャート



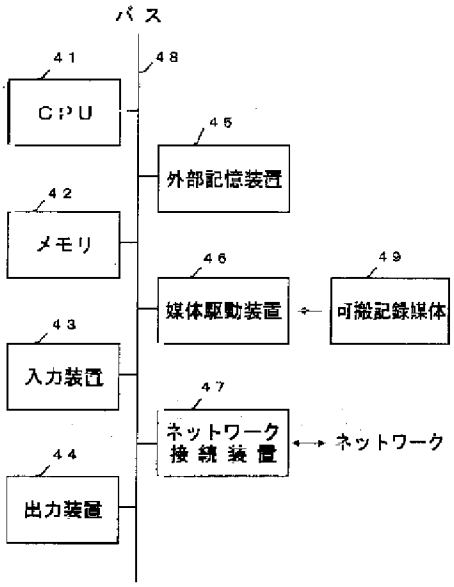

【図１３】

第４のデータ生成処理のフローチャート




【図１４】

情報処理装置の構成図



【図15】

記録媒体を示す図